

FIG. 1

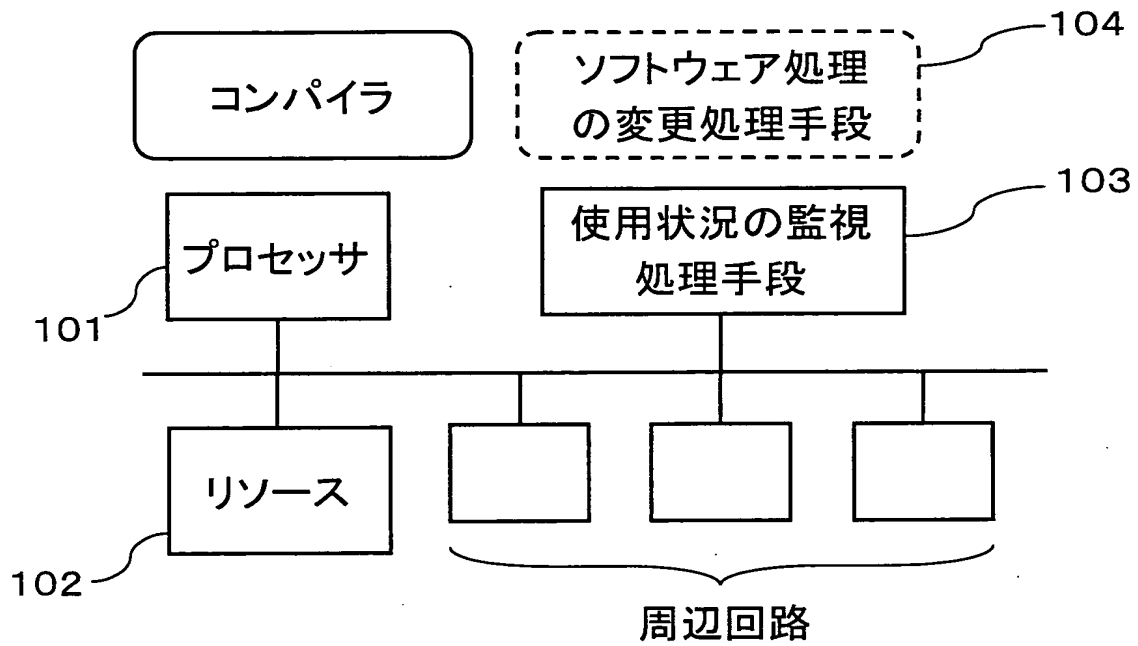


FIG. 2

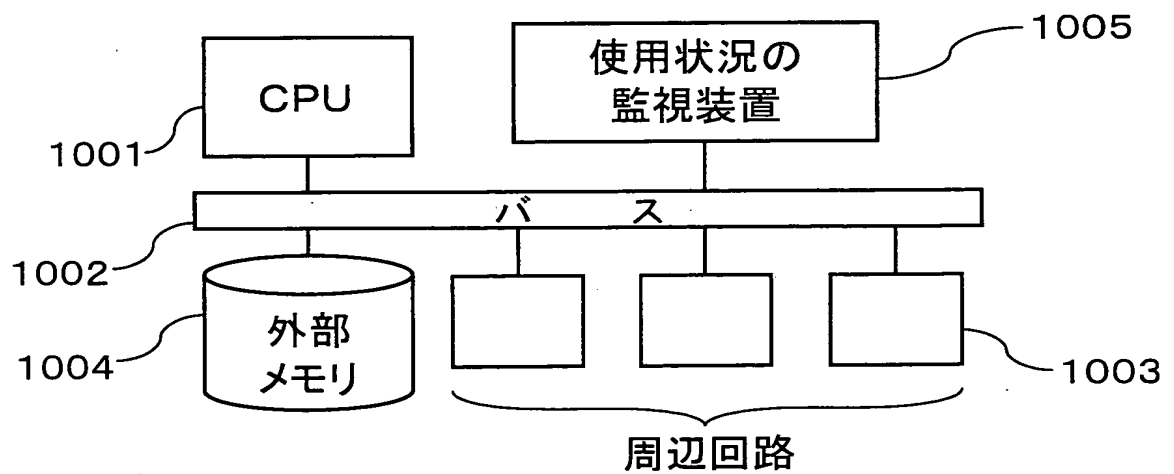


FIG. 3

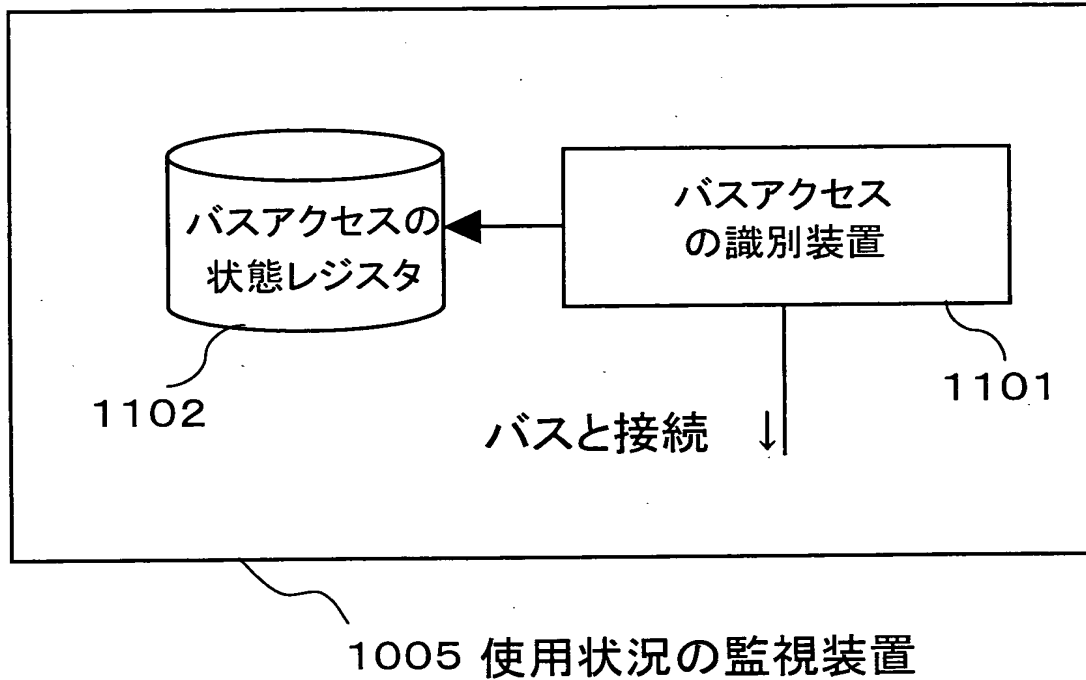


FIG. 4A

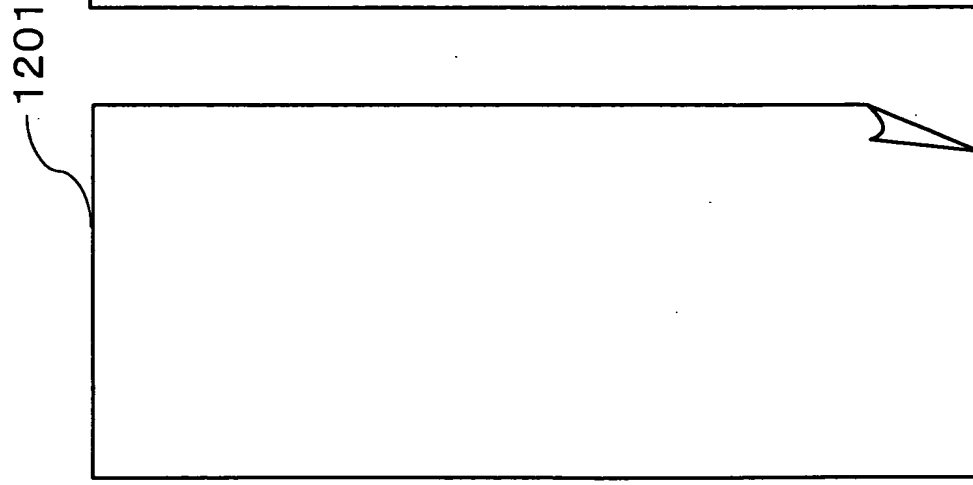


FIG. 4B

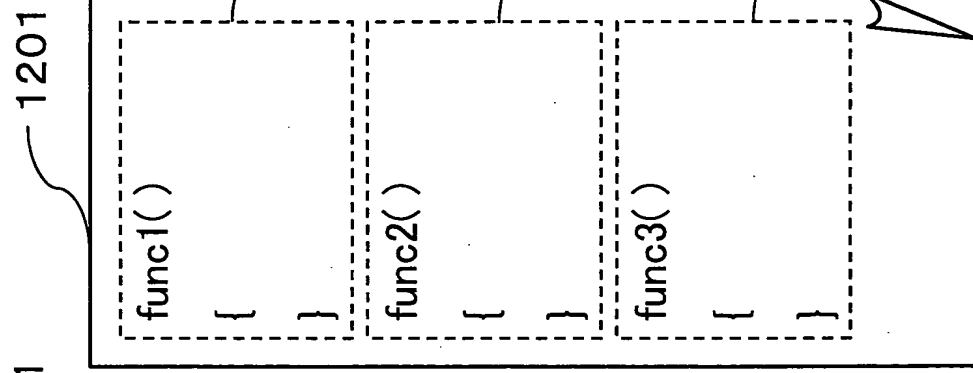


FIG. 4C

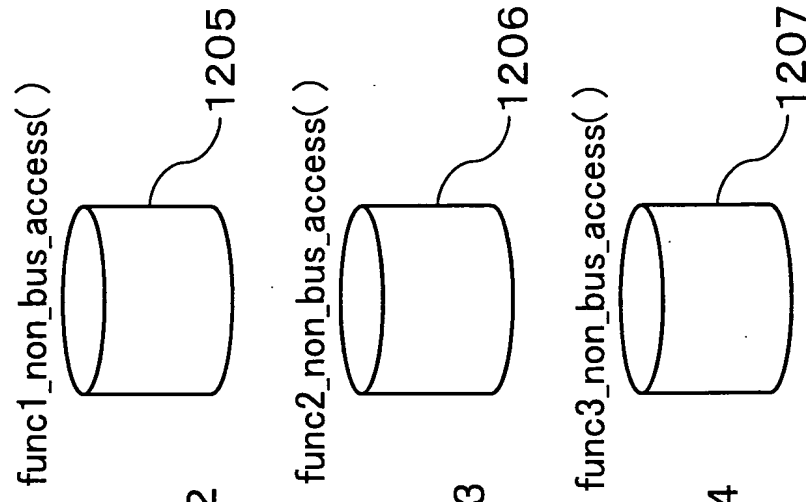


FIG. 5

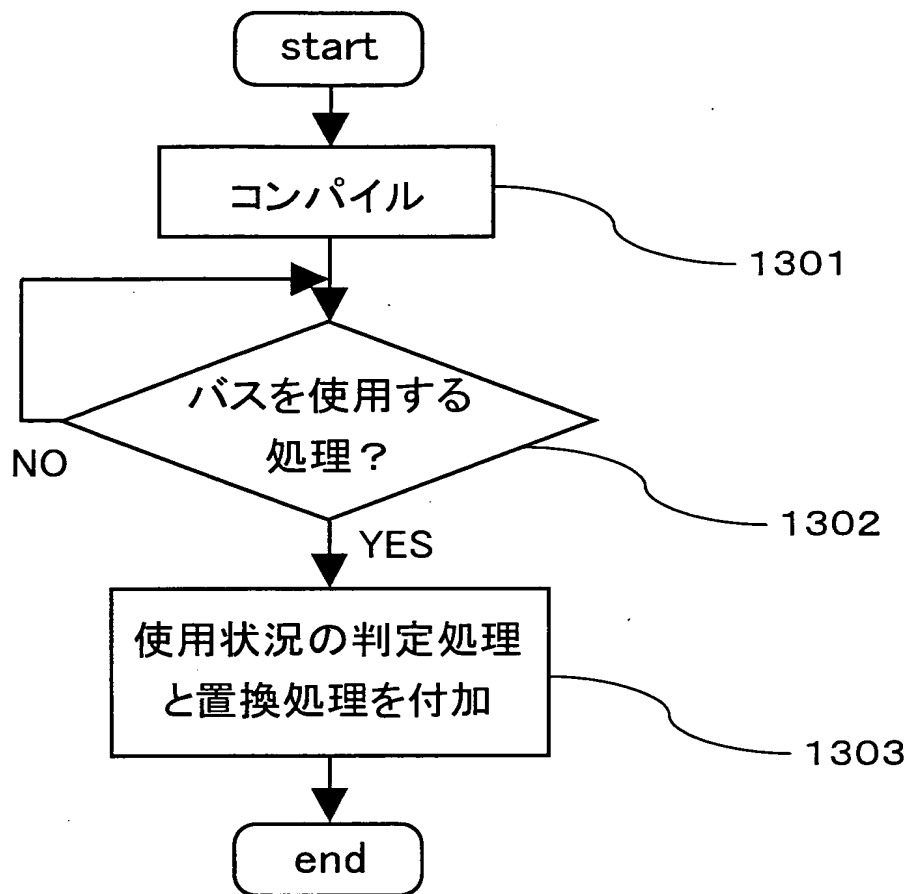


FIG. 6

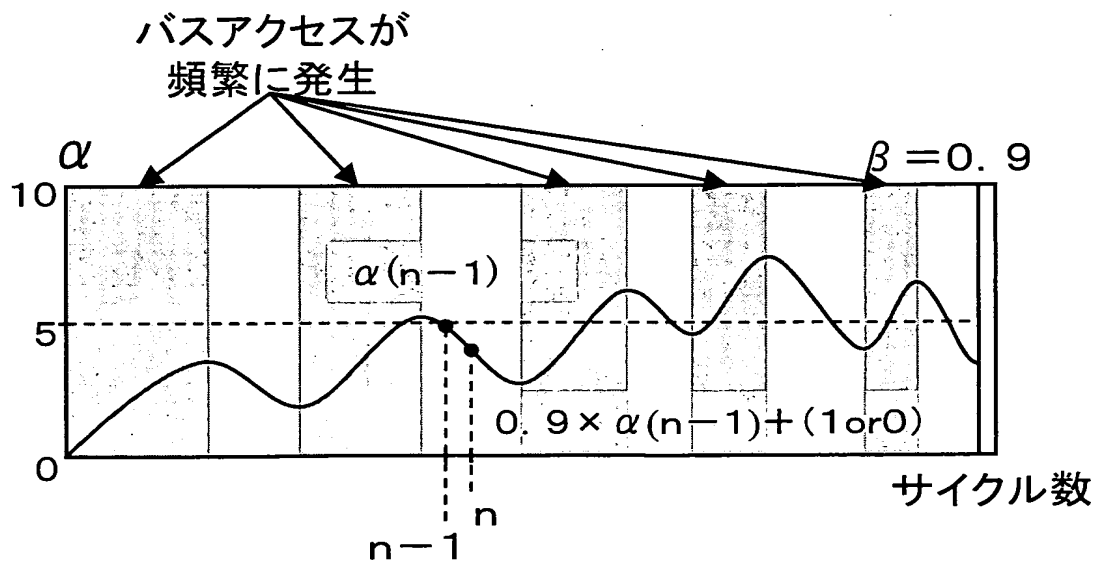


FIG. 7A

```
main()  
{  
:  
:  
  Func1()  
:  
  Func2()  
:  
  func3()  
:  
:  
}  
:  
:
```

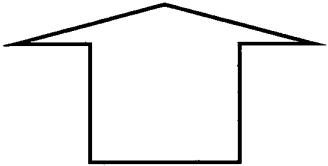


FIG. 7B

```
Func1()  
{  
  if (バスアクセスが起きている)  
    func1_non_bus_access();  
  else  
    func1();  
}  
Func2()  
{  
  if (バスアクセスが起きている)  
    func2_non_bus_access();  
  else  
    func2();  
}  
:
```

FIG. 8

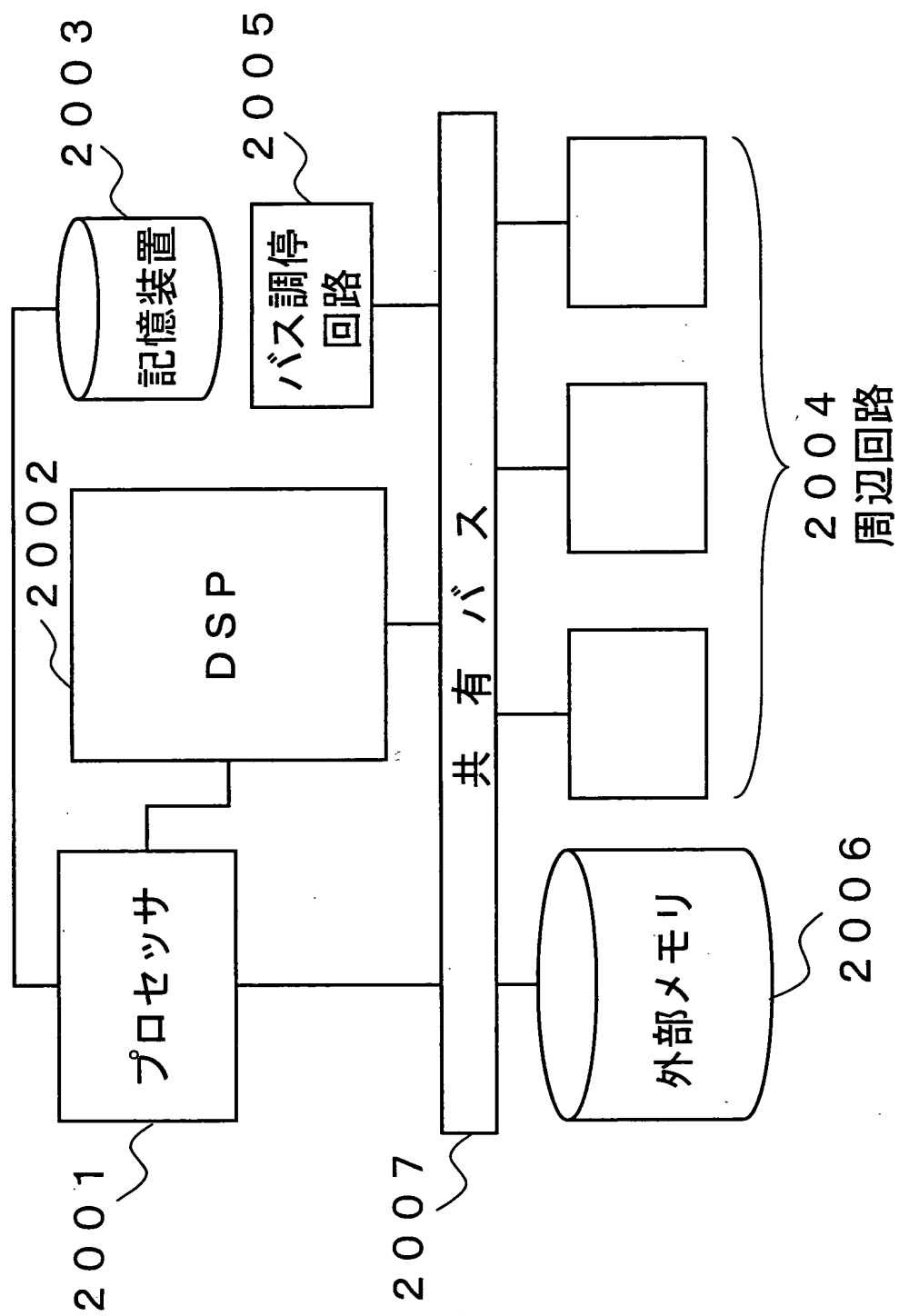


FIG. 9

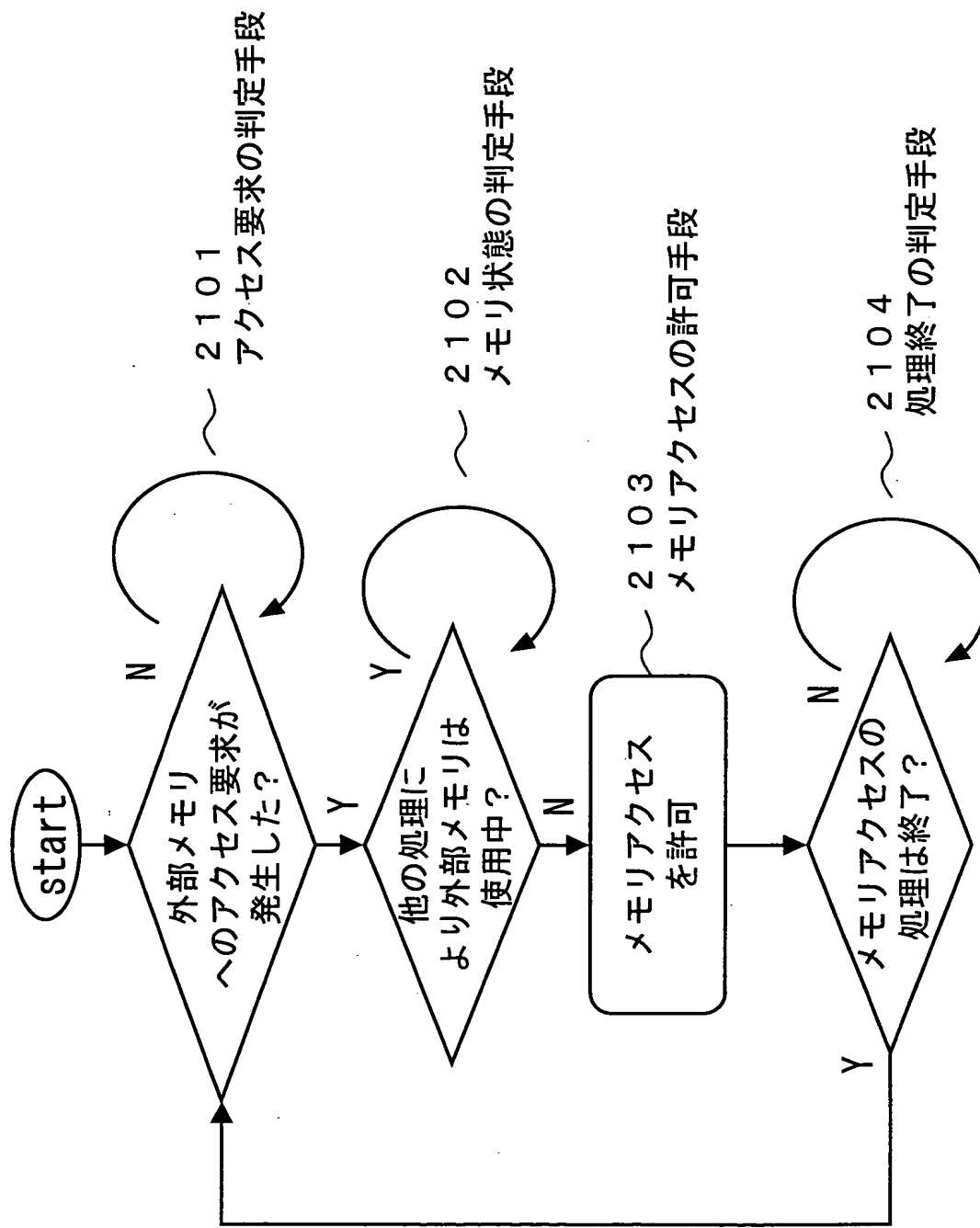


FIG. 10

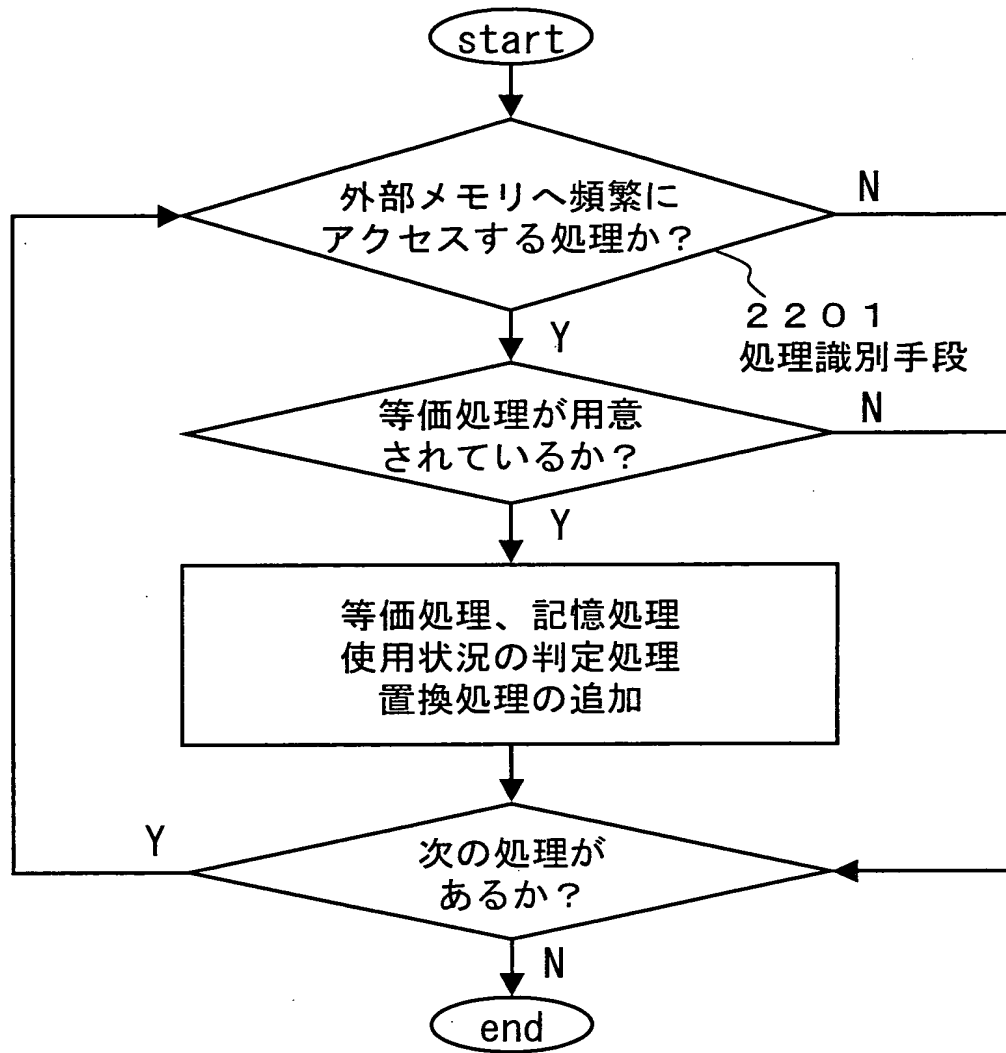




FIG. 11A

```
1: proc_task1() {  
2:     mem_acs_many();  
3: };
```

コンパイラ実行前

FIG. 11B

```
1: proc_task1() {  
2:     if( task_time < DEFINED_TIME ) {  
3:         start_time = timer_count;  
4:         mem_acs_many();  
5:         end_time   = timer_count;  
6:         task_time = end_time - start_time;  
7:     }  
8:     else{  
9:         mem_acs_few();  
10:    };  
11: };
```

コンパイラ実行後

FIG. 12

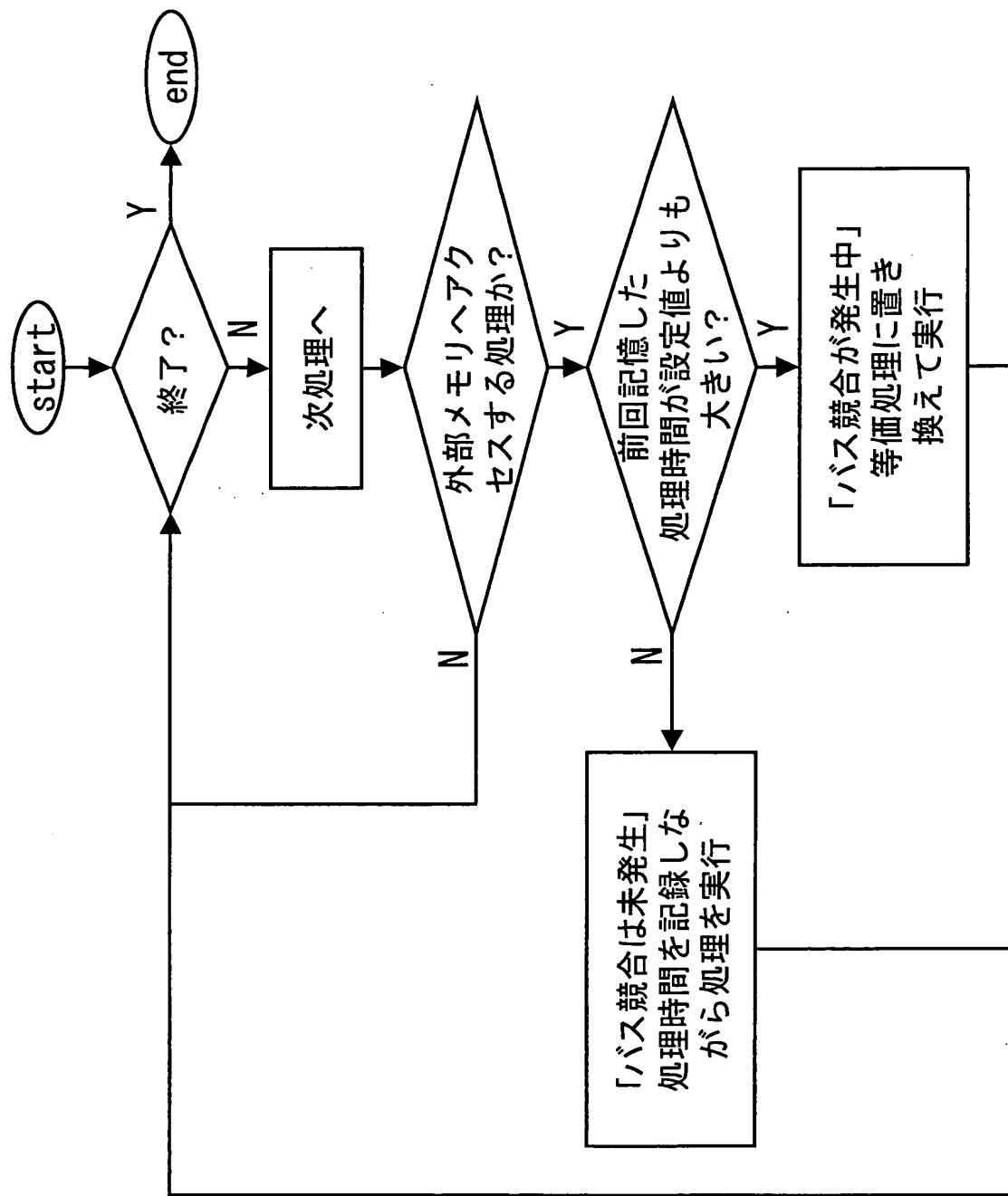


FIG. 13

処理時間 —— 設定値	等価処理へ置き 換える確率[%]	等価処理へ置き 換えない確率[%]
~ 2. 0	60	40
2. 0 ~ 1. 8	70	30
1. 8 ~ 1. 6	80	20
1. 6 ~ 1. 4	90	10
1. 4 ~ 1. 2	95	5
1. 2 ~ 1. 0	97	3
1 未満	0	100

FIG. 14

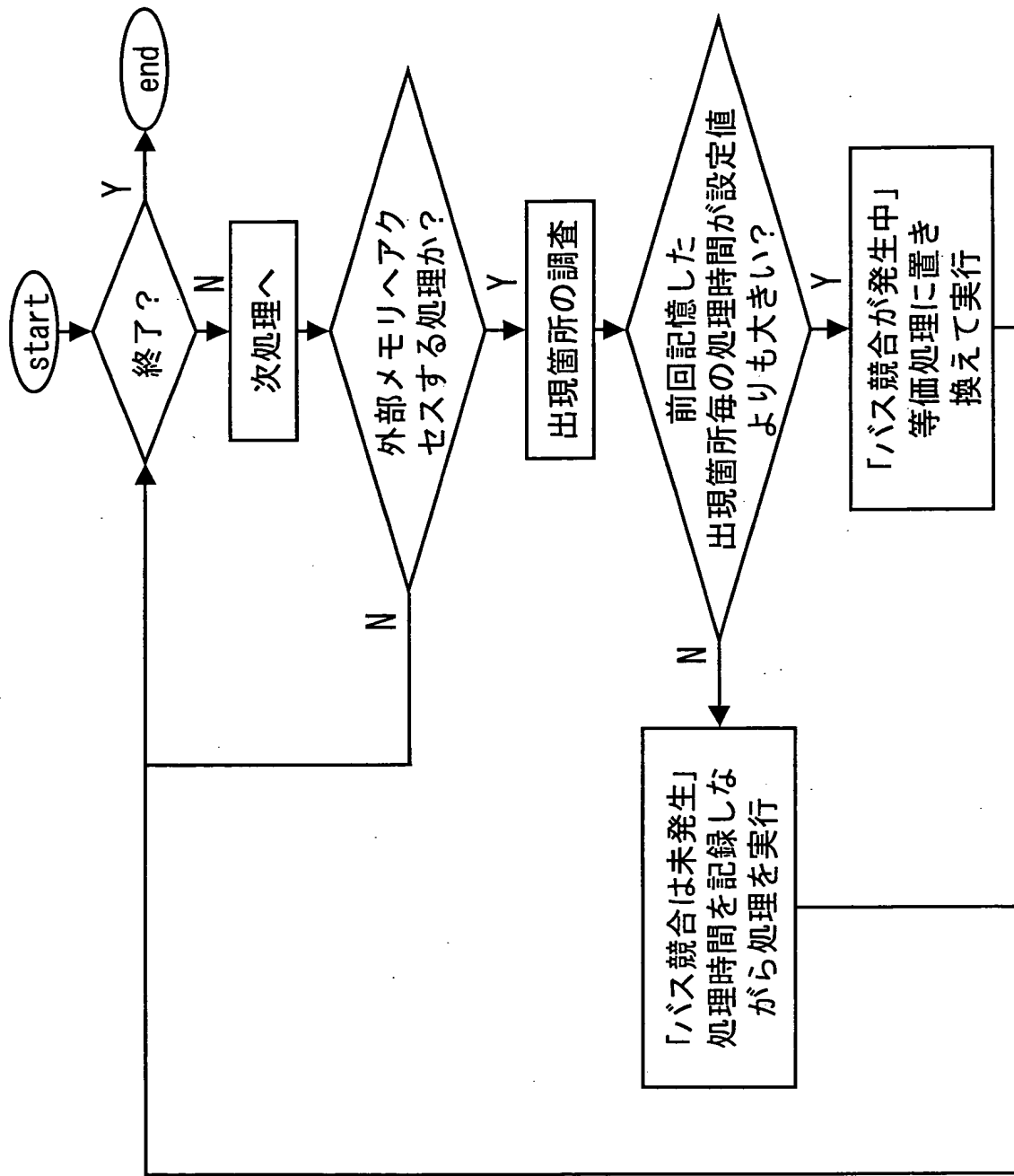


FIG. 15

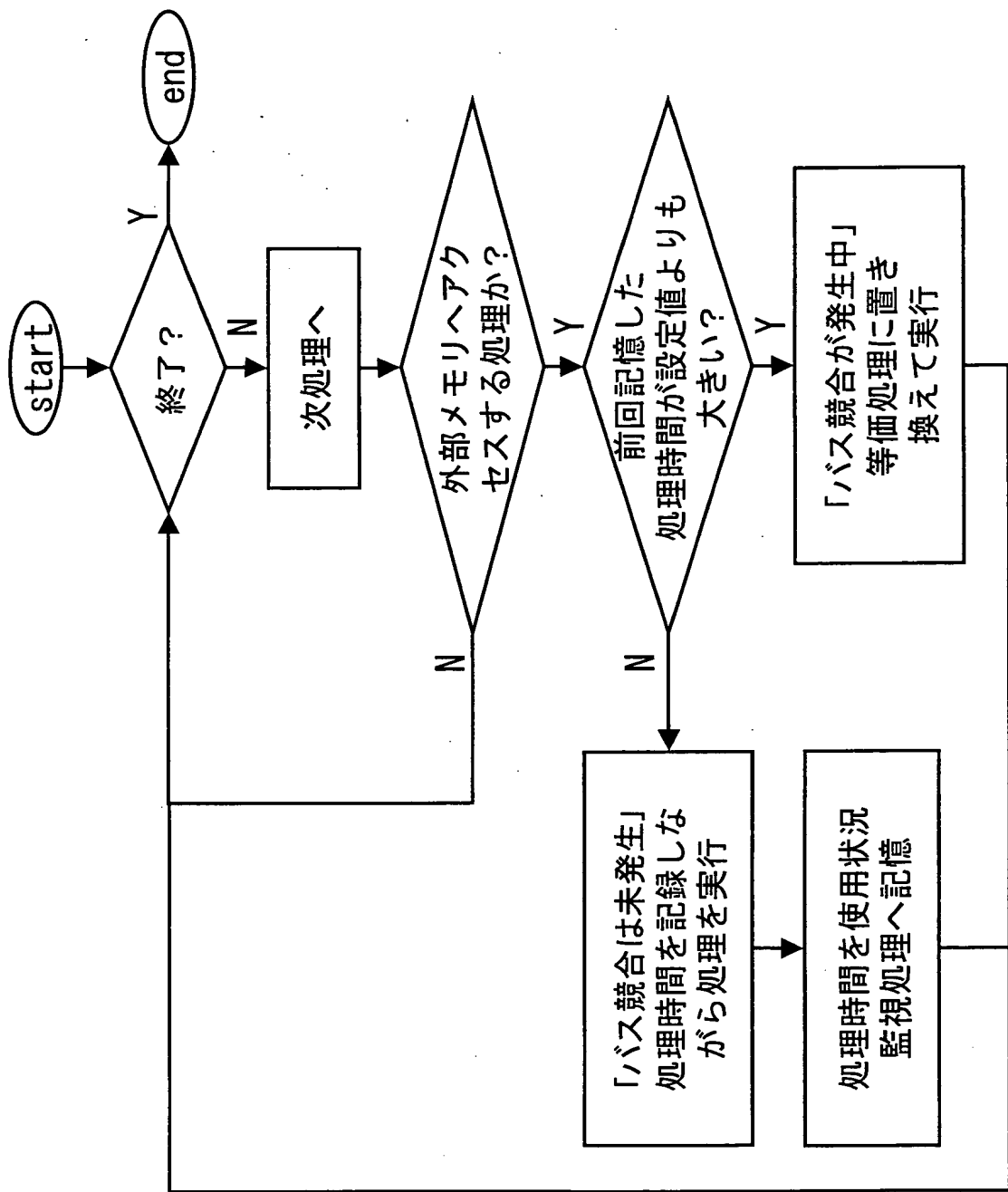


FIG. 16

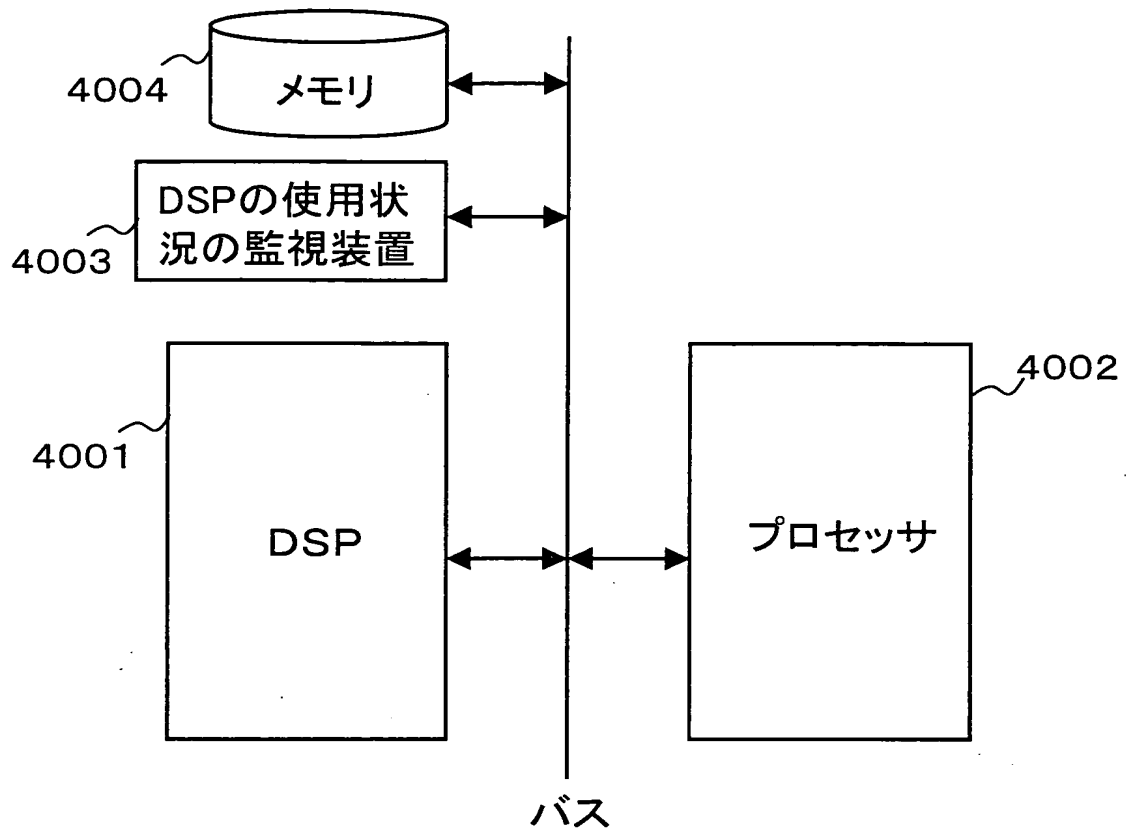


FIG. 17

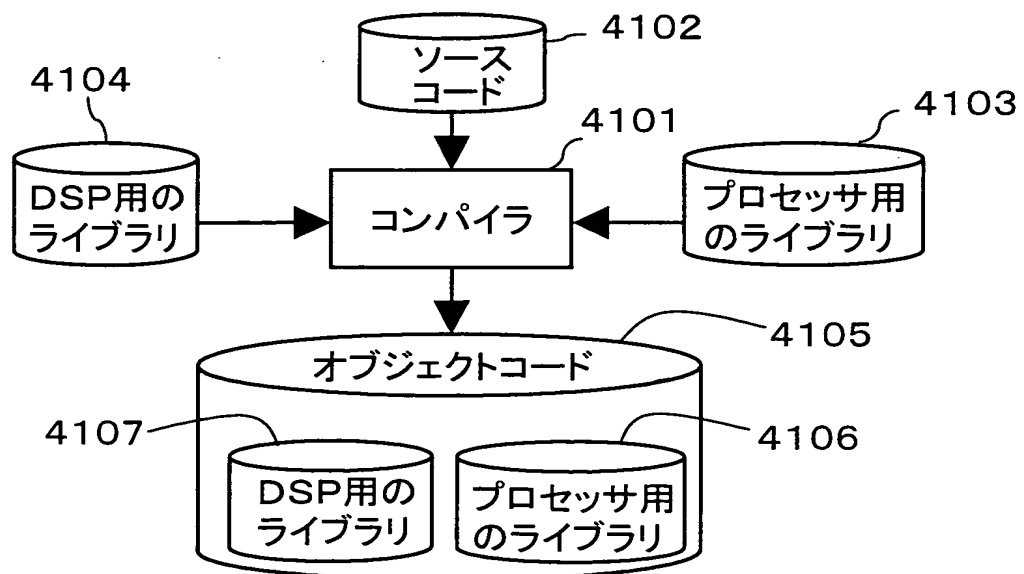


FIG. 18A

```
1: main()  
2: {  
3:   :  
4:   func1()  
5:   :  
6:   func2()  
7:   :  
8:   func3()  
9:   :  
10: }
```

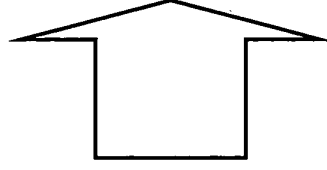


FIG. 18B

```
1: main()  
2: {  
3:   :  
4:   func1()  
5:   :  
6:   func2()  
7:   {  
8:     if( DSPが待ち状態)  
9:       func2_dsp();  
10:    else  
11:      func2_cpu();  
12:    }  
13:   :  
14:   func3()  
15:   :  
16: }
```

FIG. 19

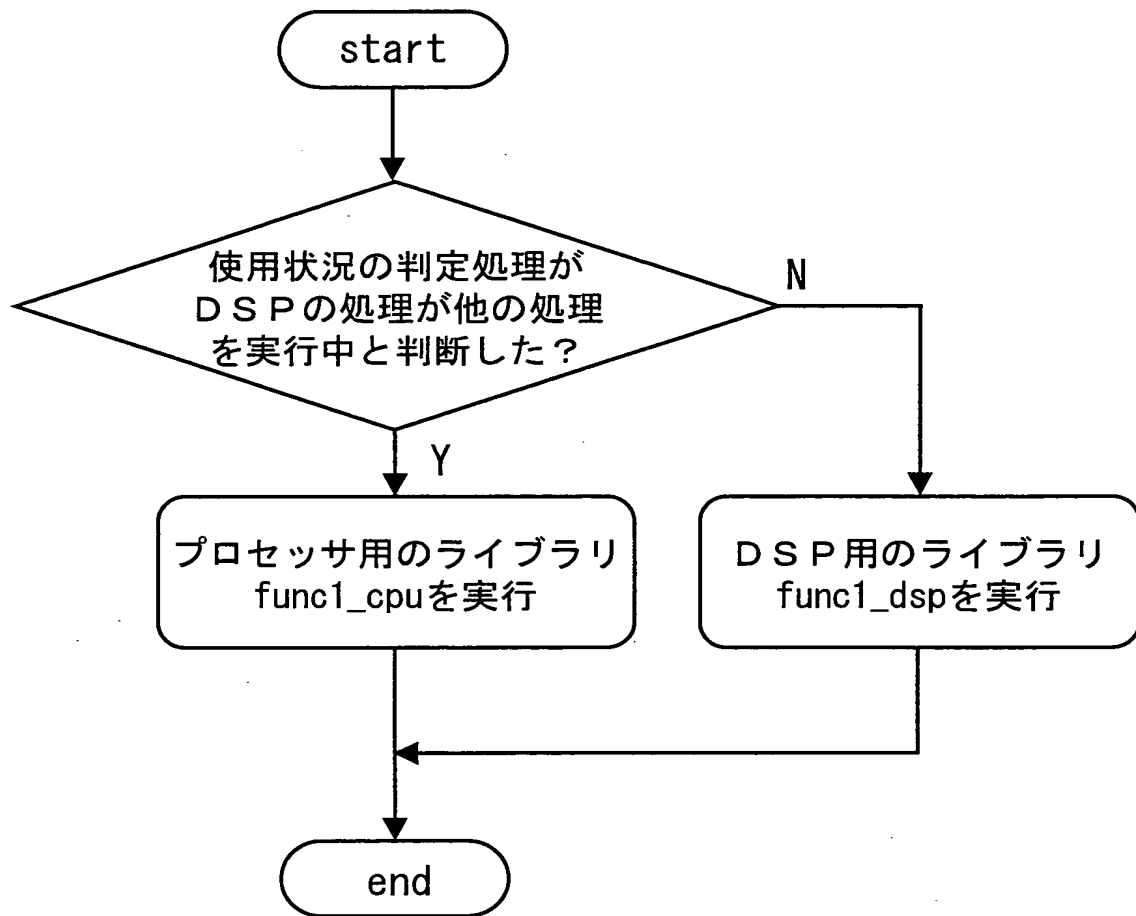




FIG. 20A

DSPのタスクが競合しない場合

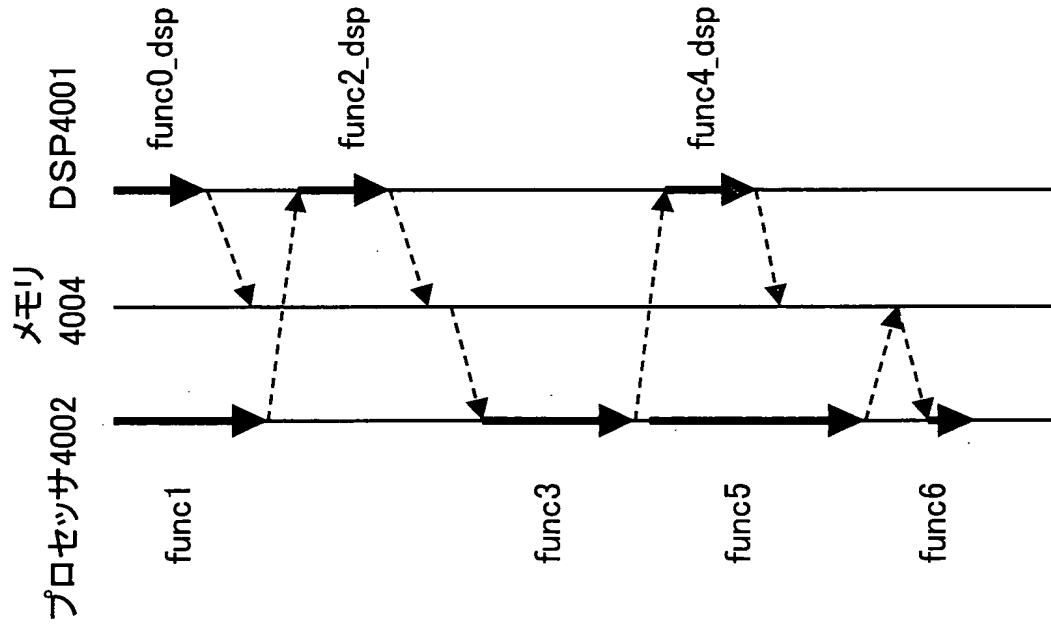


FIG. 20B

DSPのタスクが競合した場合

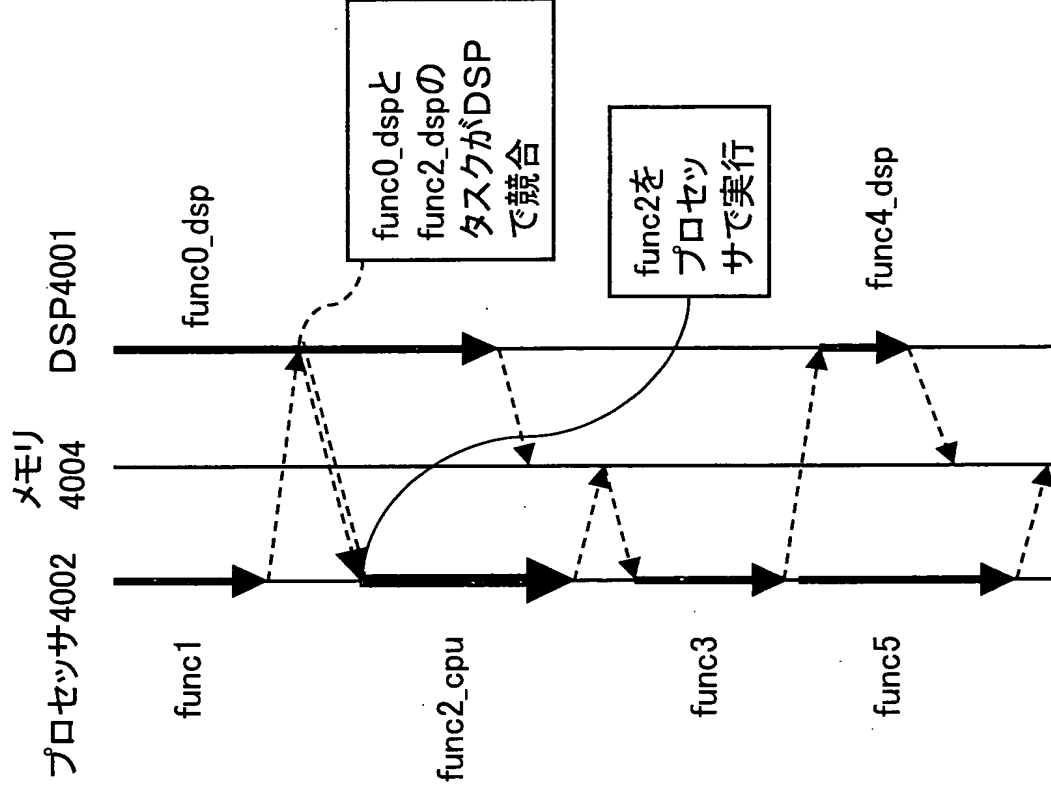


FIG. 21

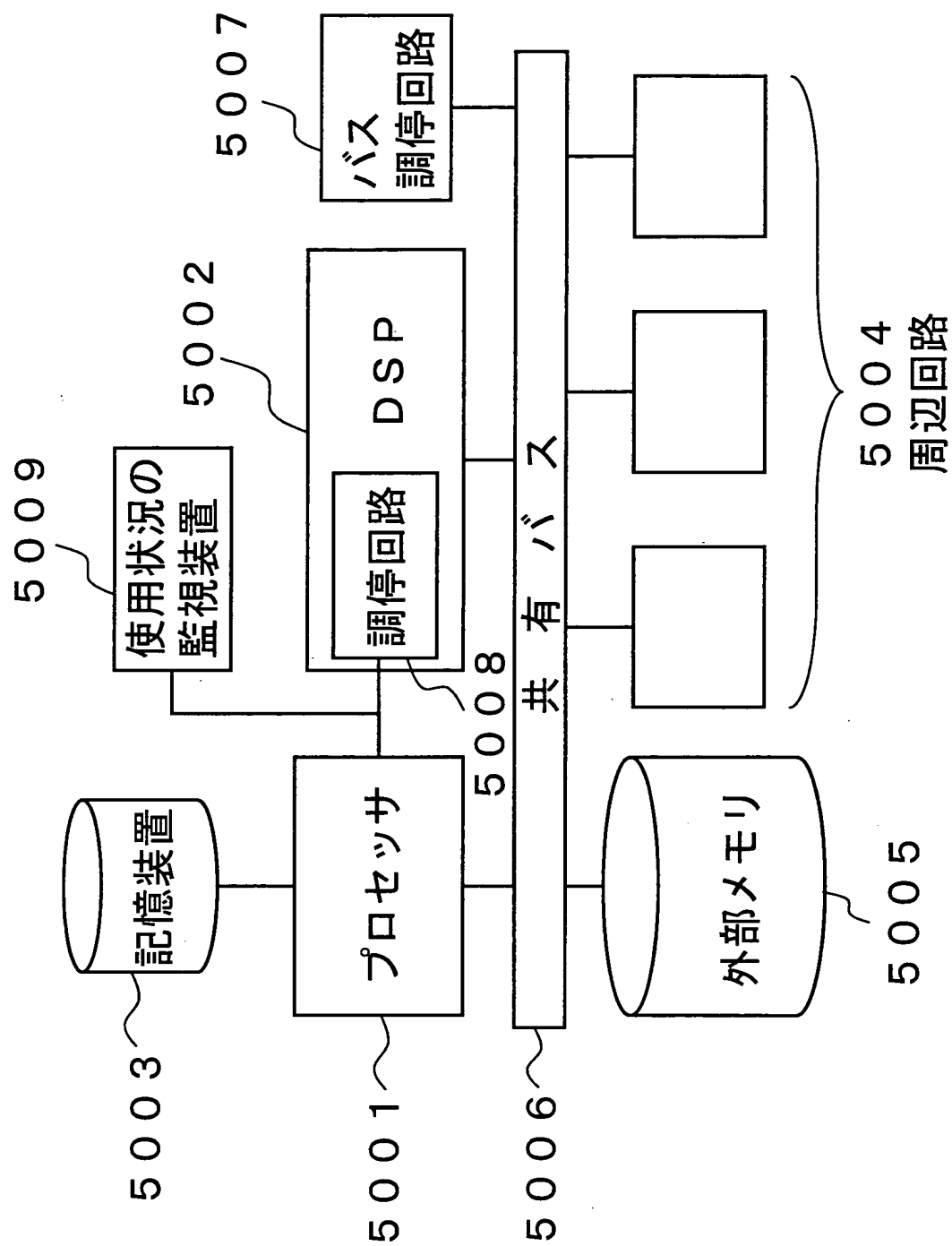


FIG. 22

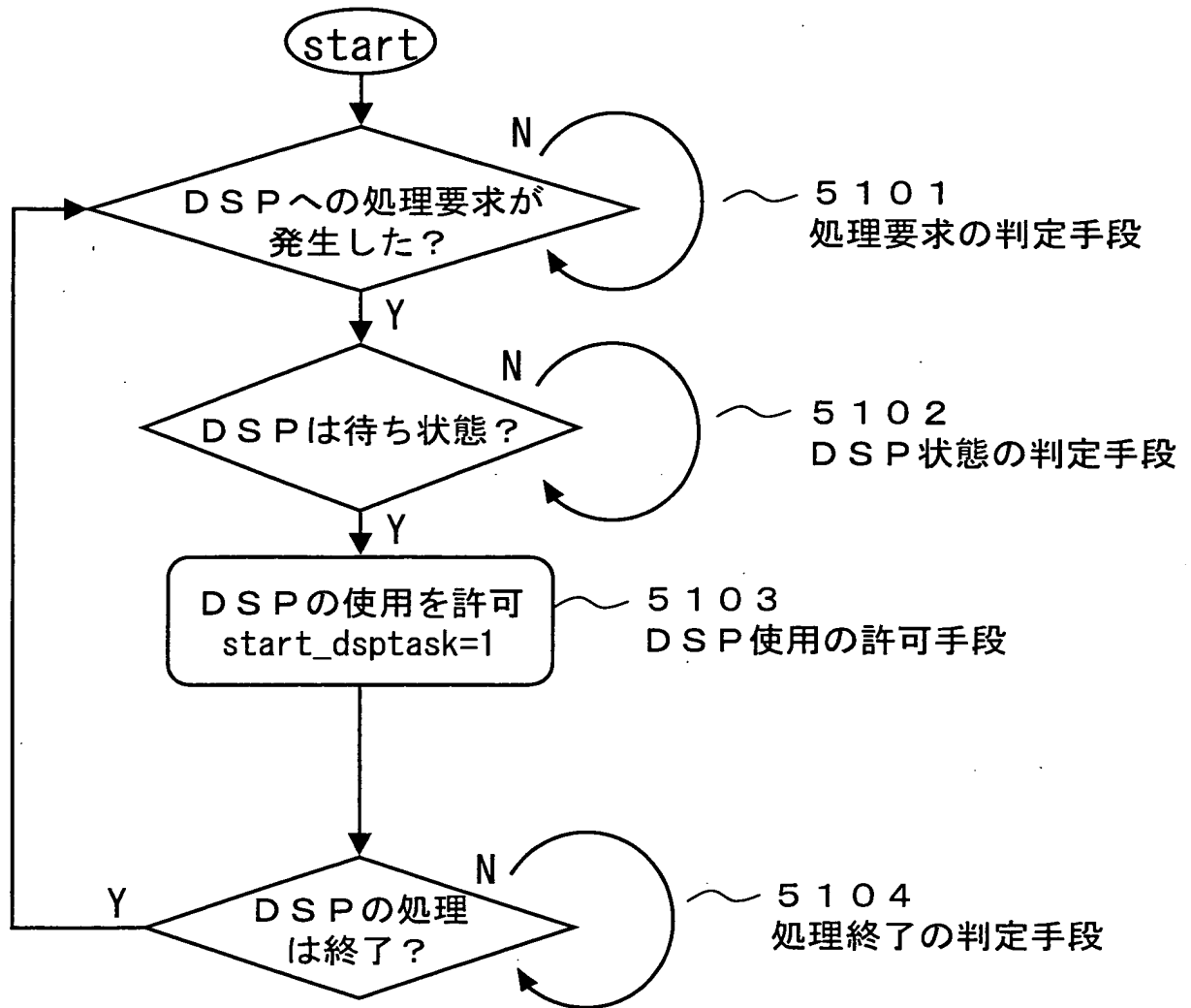


FIG. 23

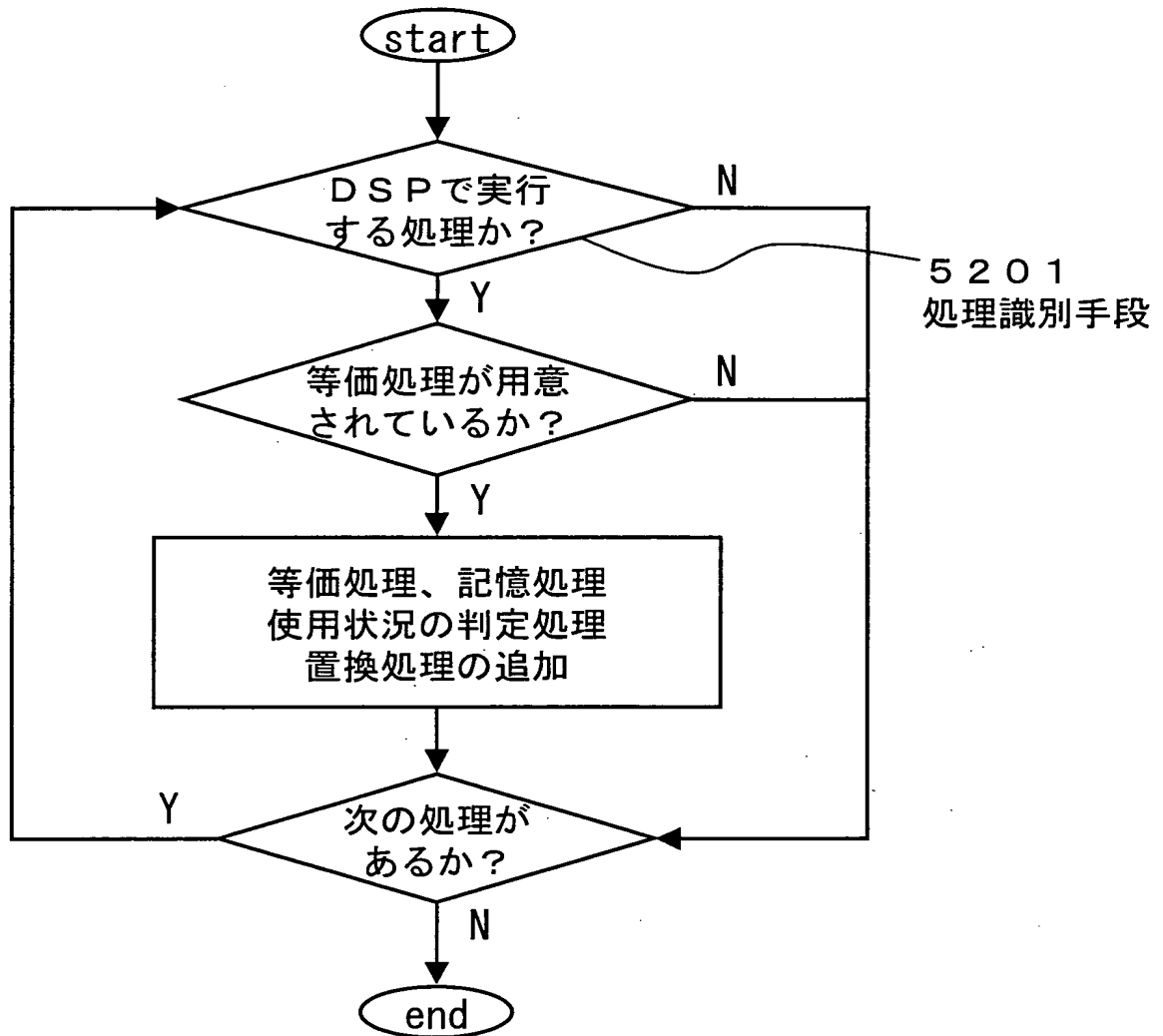


FIG. 24A

```
1: proc_task1() {  
2:     dsp_task();  
3: };
```

コンパイラ実行前

FIG. 24B

```
1: proc_task1() {  
2:     if( wait_time < DEFINED_TIME ) {  
3:         start_time = timer_count;  
4:         (DSPが処理を開始できるまで待つ)  
5:         end_time   = timer_count;  
6:         wait_time = end_time - start_time;  
7:         dsp_task();  
8:     }  
9:     else{  
10:        proc_dsptask();  
11:    };  
12: };
```

コンパイラ実行後

FIG. 25

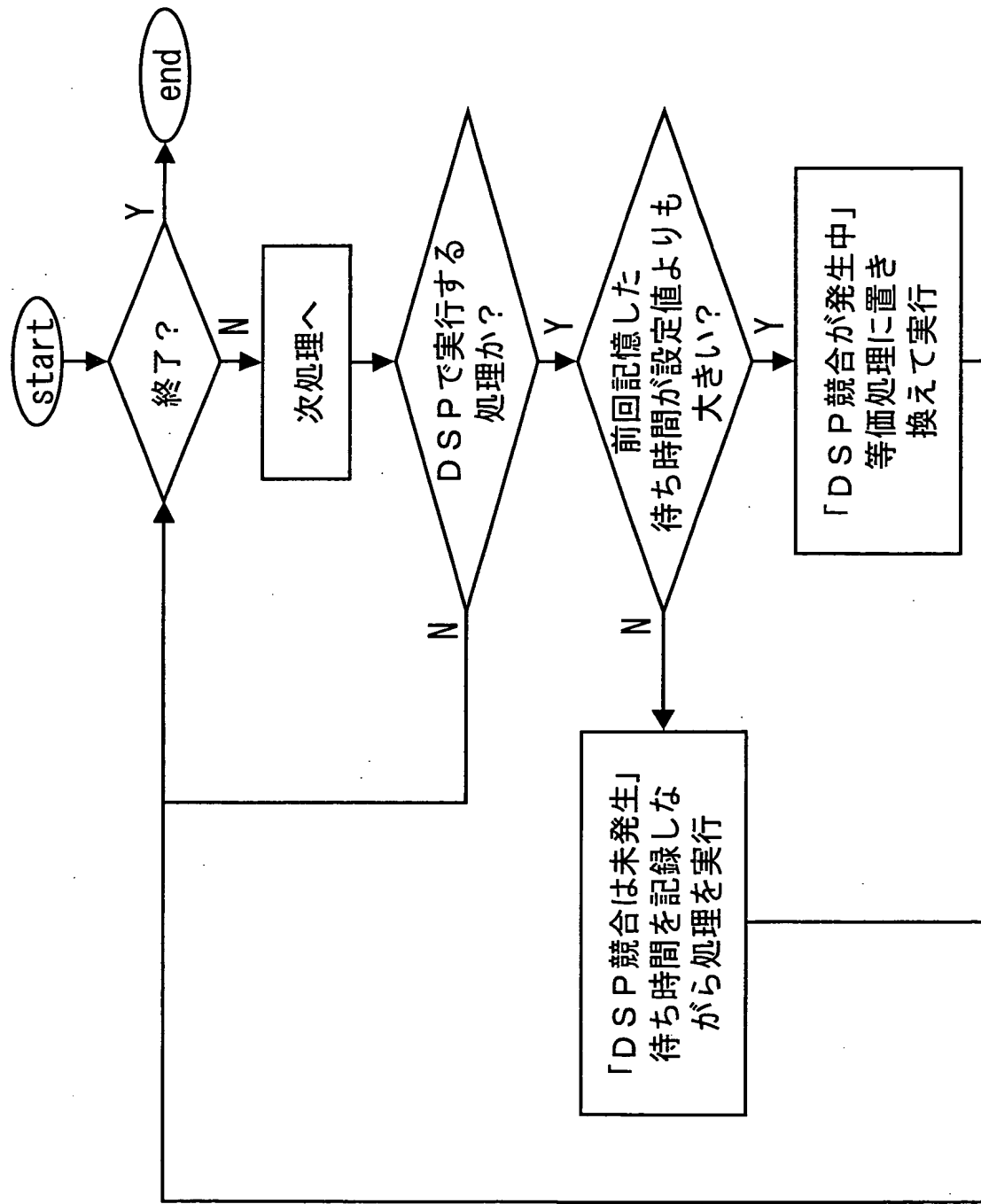


FIG. 26

待ち時間 設定値	等価処理へ置き 換える確率[%]	等価処理へ置き 換えない確率[%]
～2. 0	60	40
2. 0～1. 8	70	30
1. 8～1. 6	80	20
1. 6～1. 4	90	10
1. 4～1. 2	95	5
1. 2～1. 0	97	3
1未満	0	100

FIG. 27

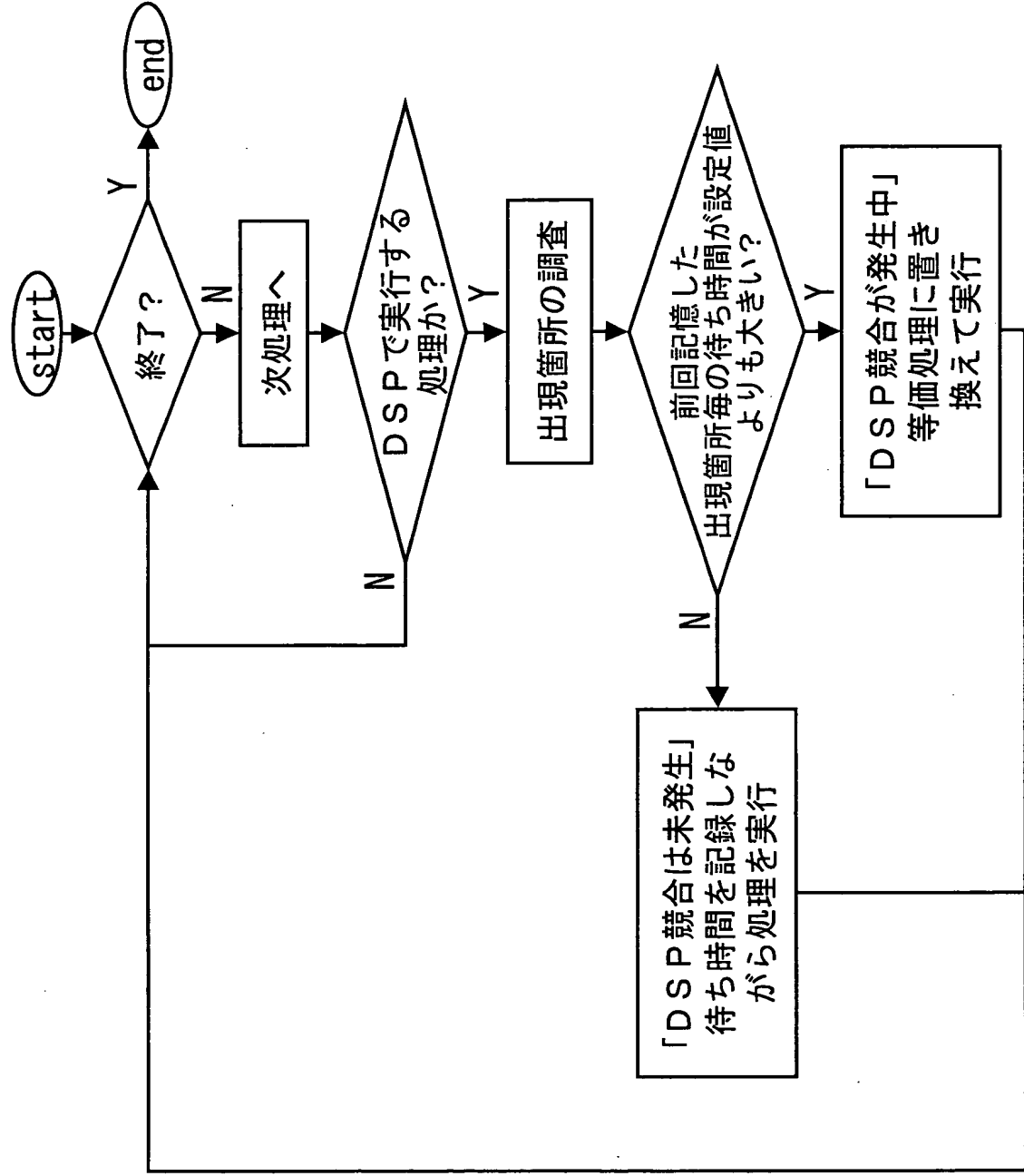




FIG. 28

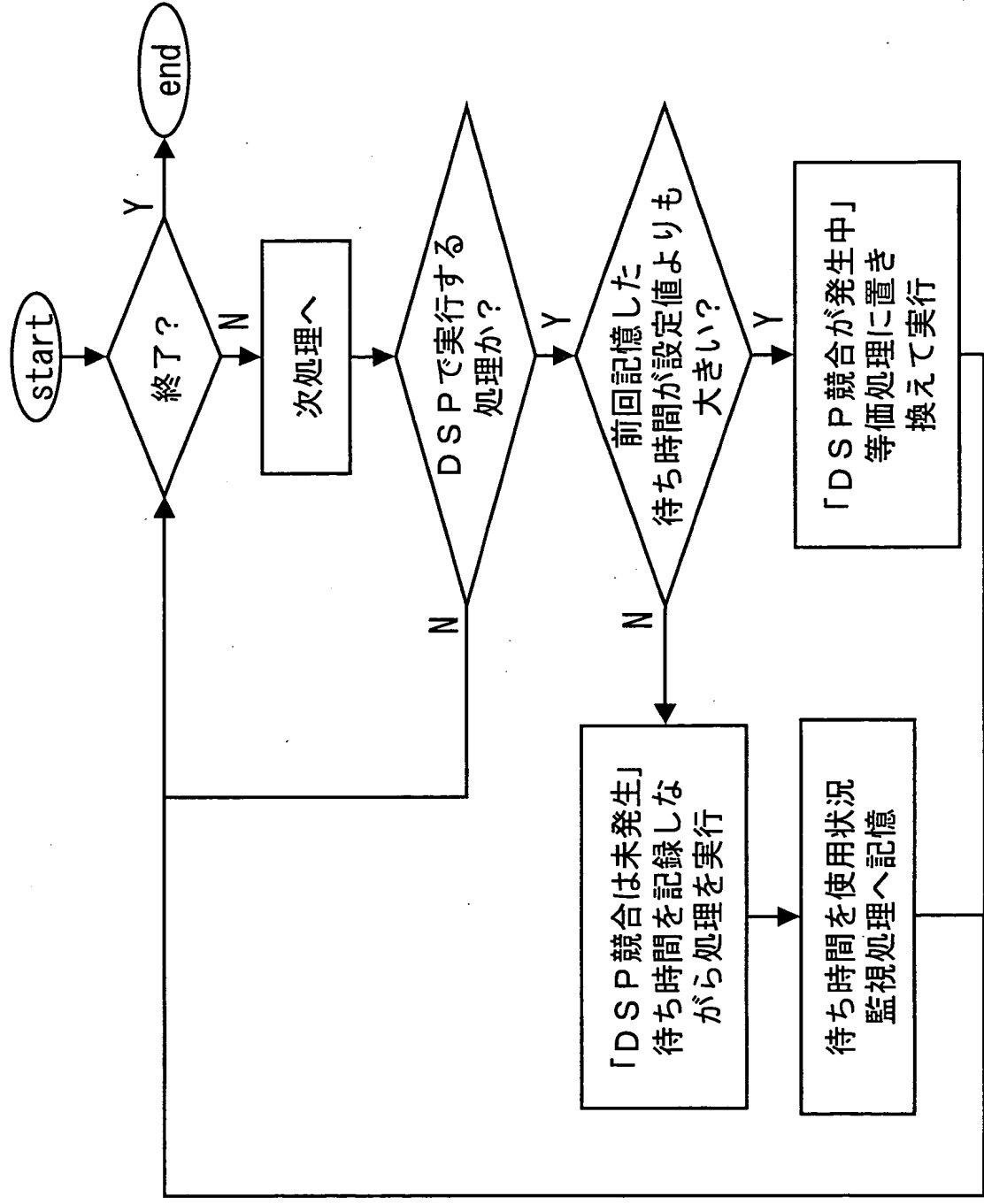


FIG. 29

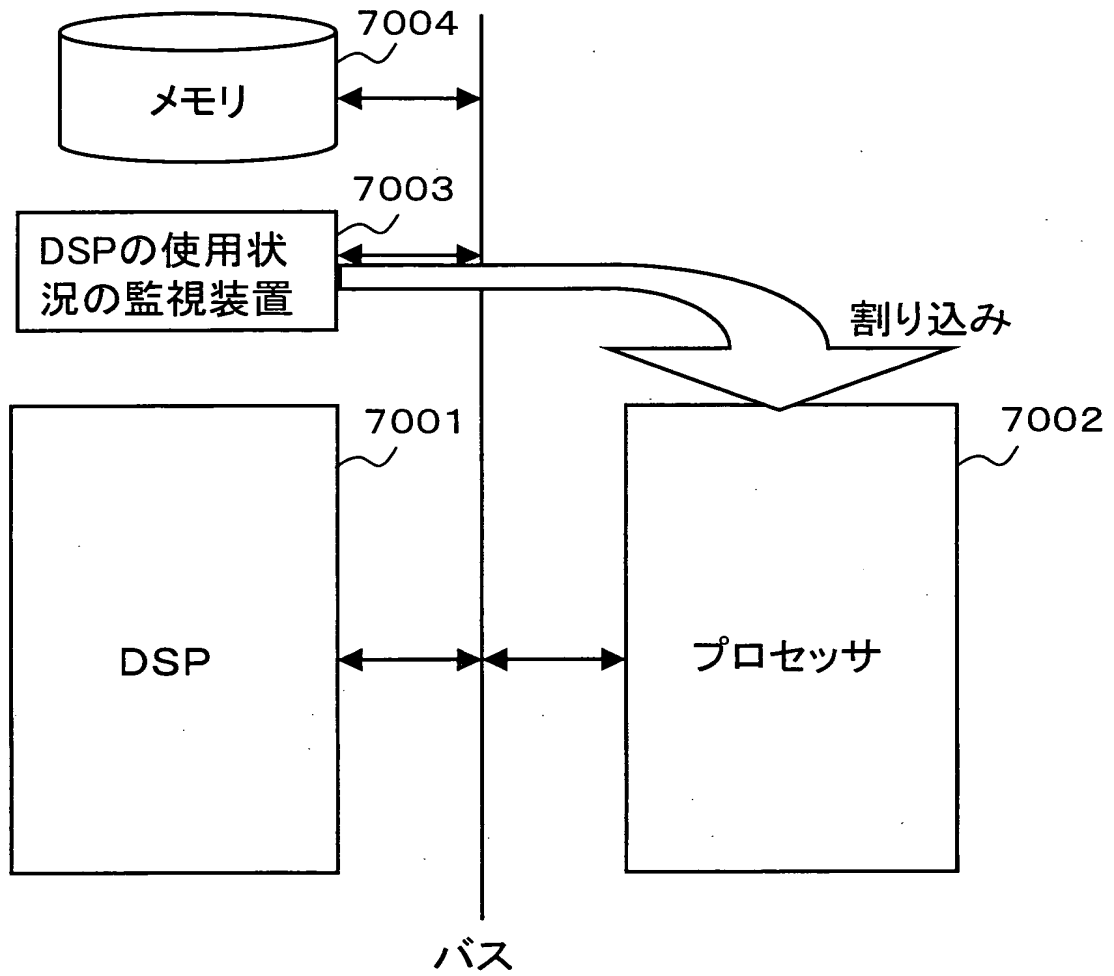


FIG. 30

```
1:  main( )  
2:  {  
3:      :  
4:      func1()  
5:      func2()  
6:      func3()  
7:      func4()  
8:      func5()  
9:      func6()  
10:     func7()  
11:     func8()  
12:     :  
13: }
```

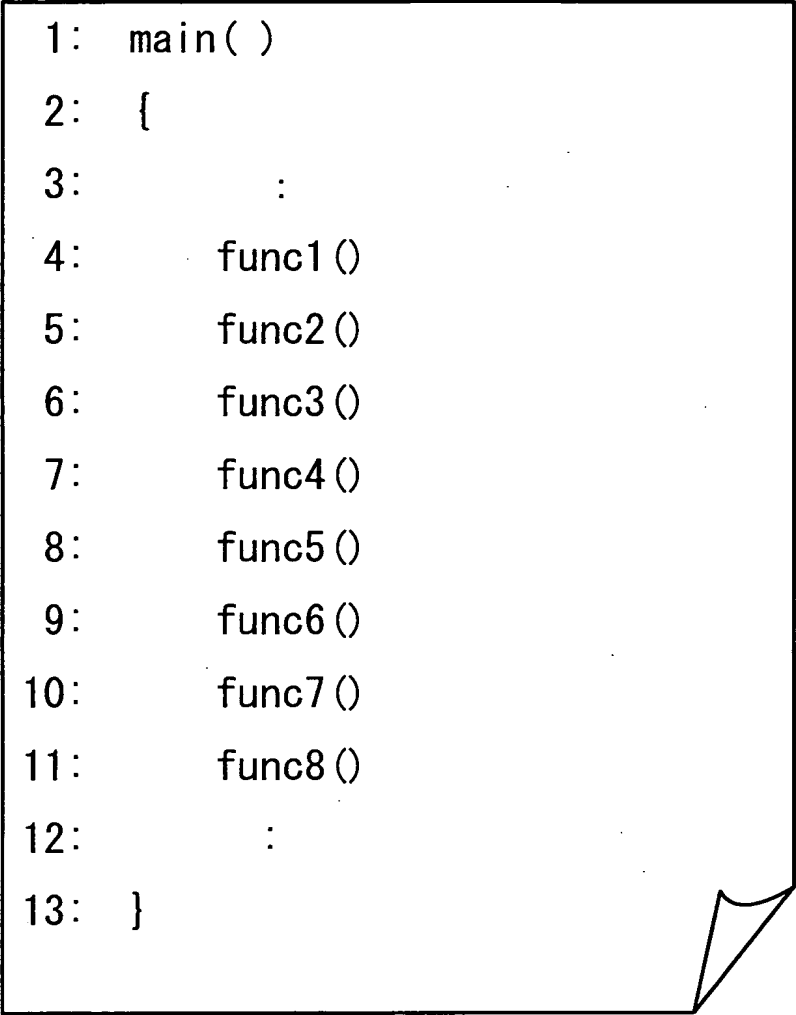


FIG. 31

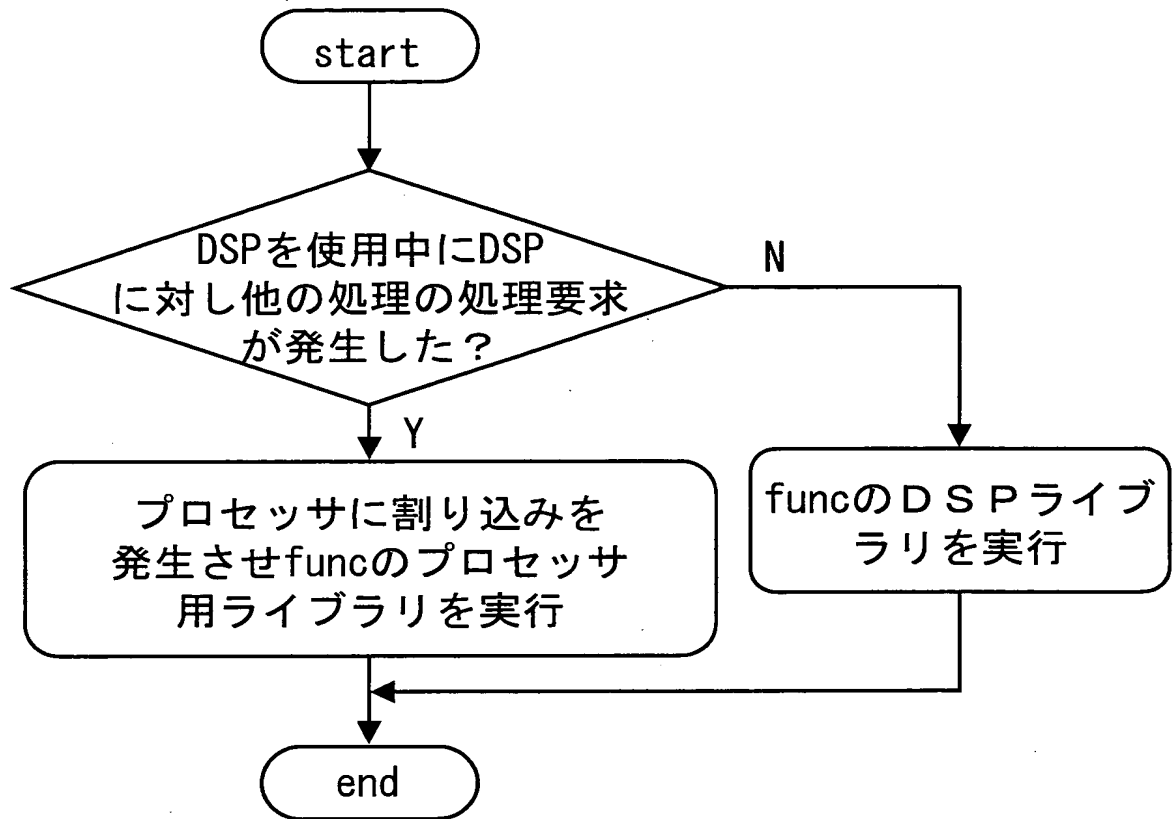


FIG. 32A

DSPのタスクが競合しない場合

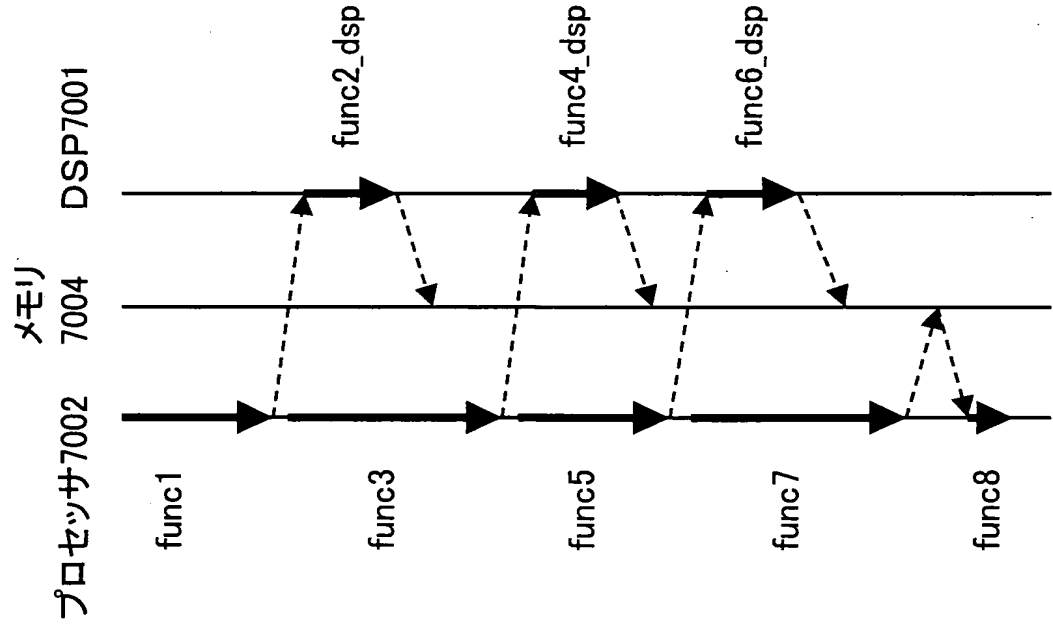


FIG. 32B

DSPのタスクが競合した場合

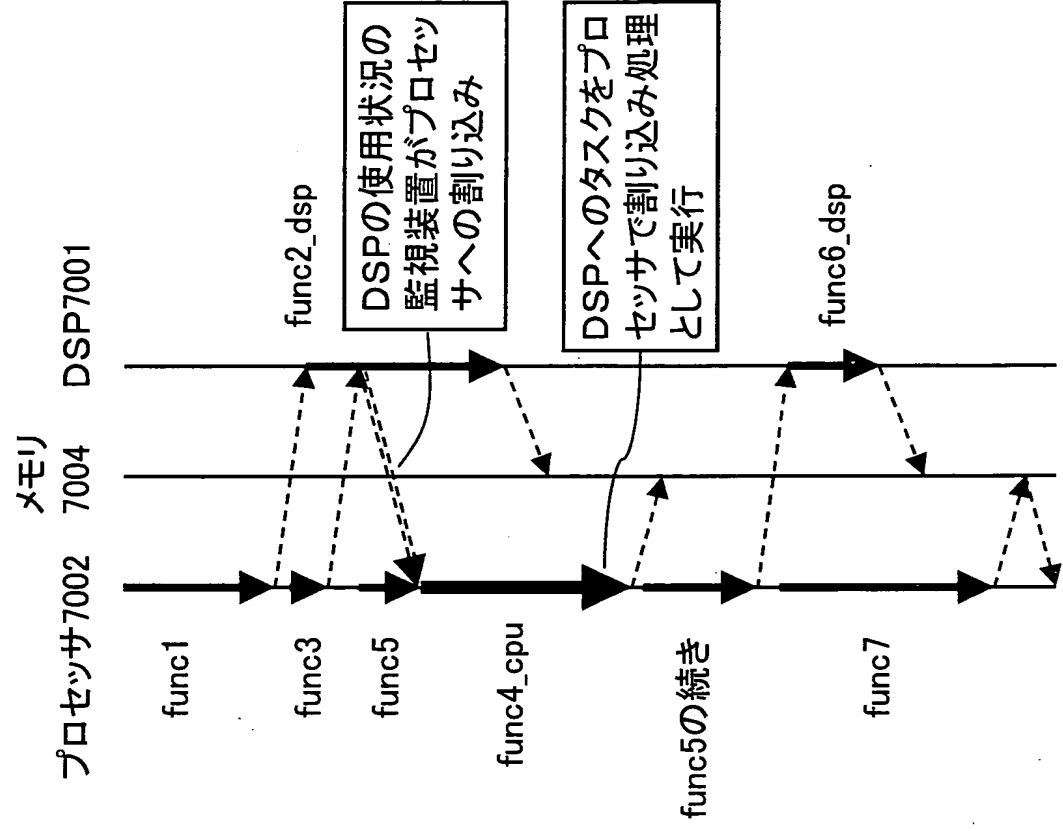


FIG. 33

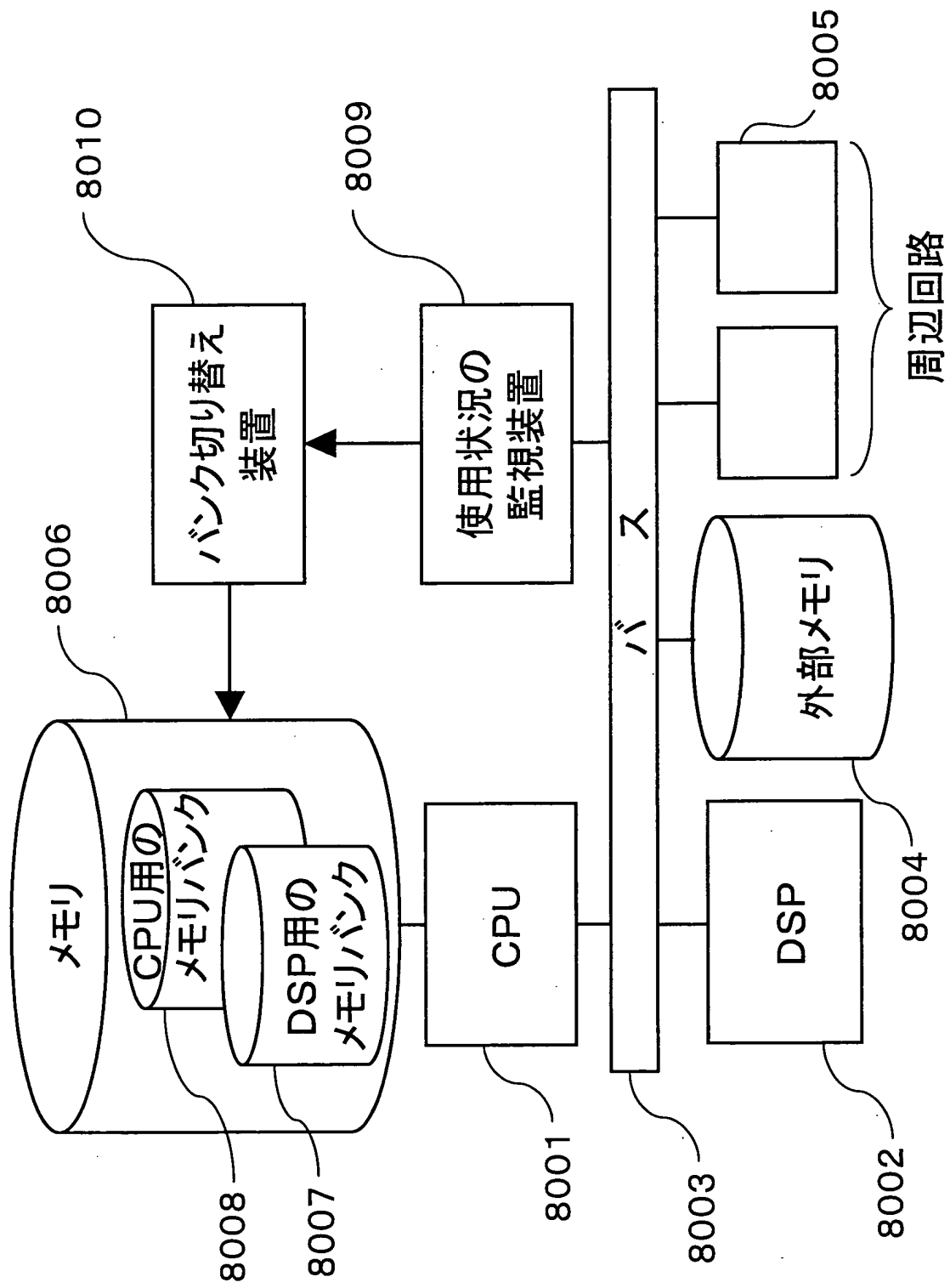


FIG. 34A

プログラム 8101

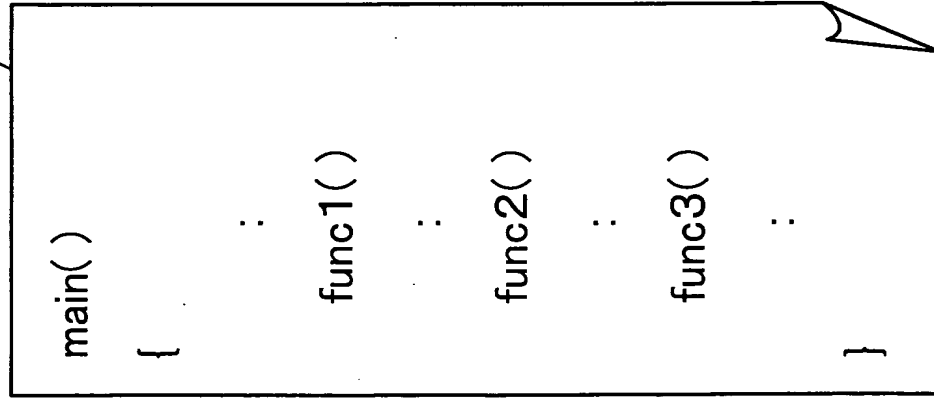


FIG. 34B

DSP使用ライブラリ

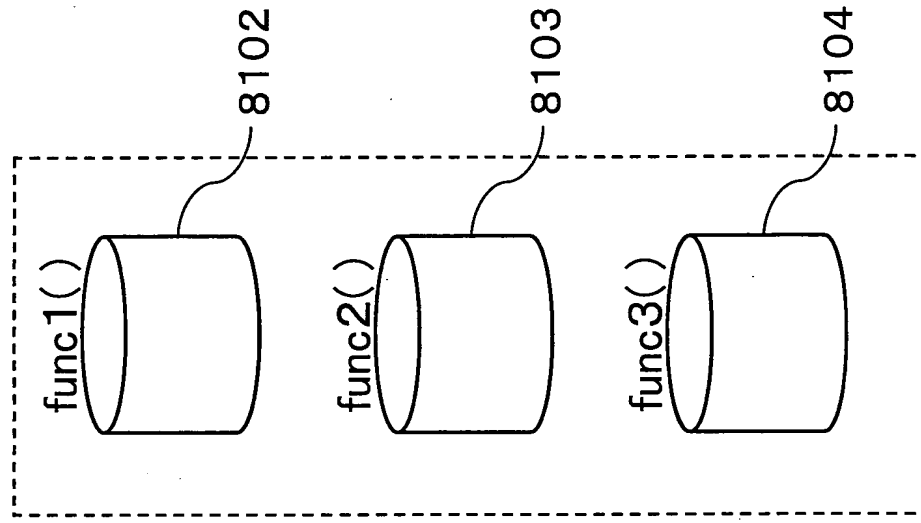


FIG. 34C

CPU使用ライブラリ

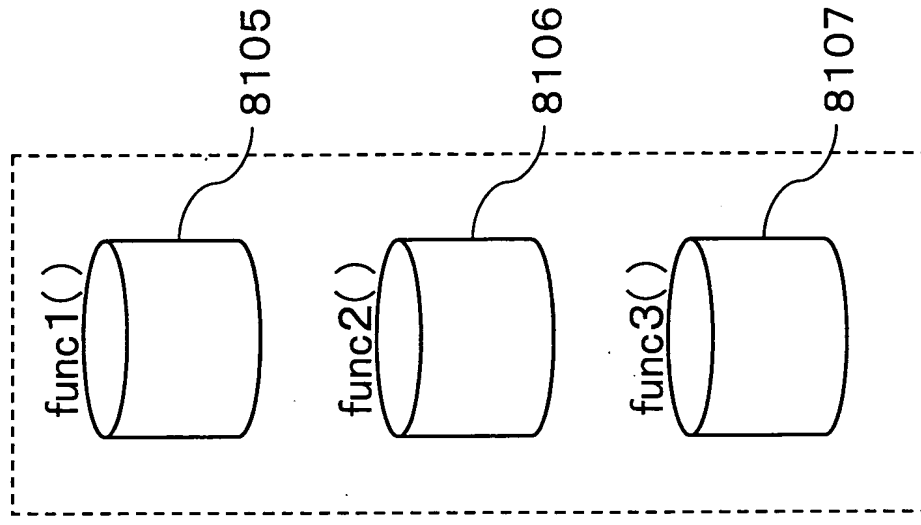


FIG. 35

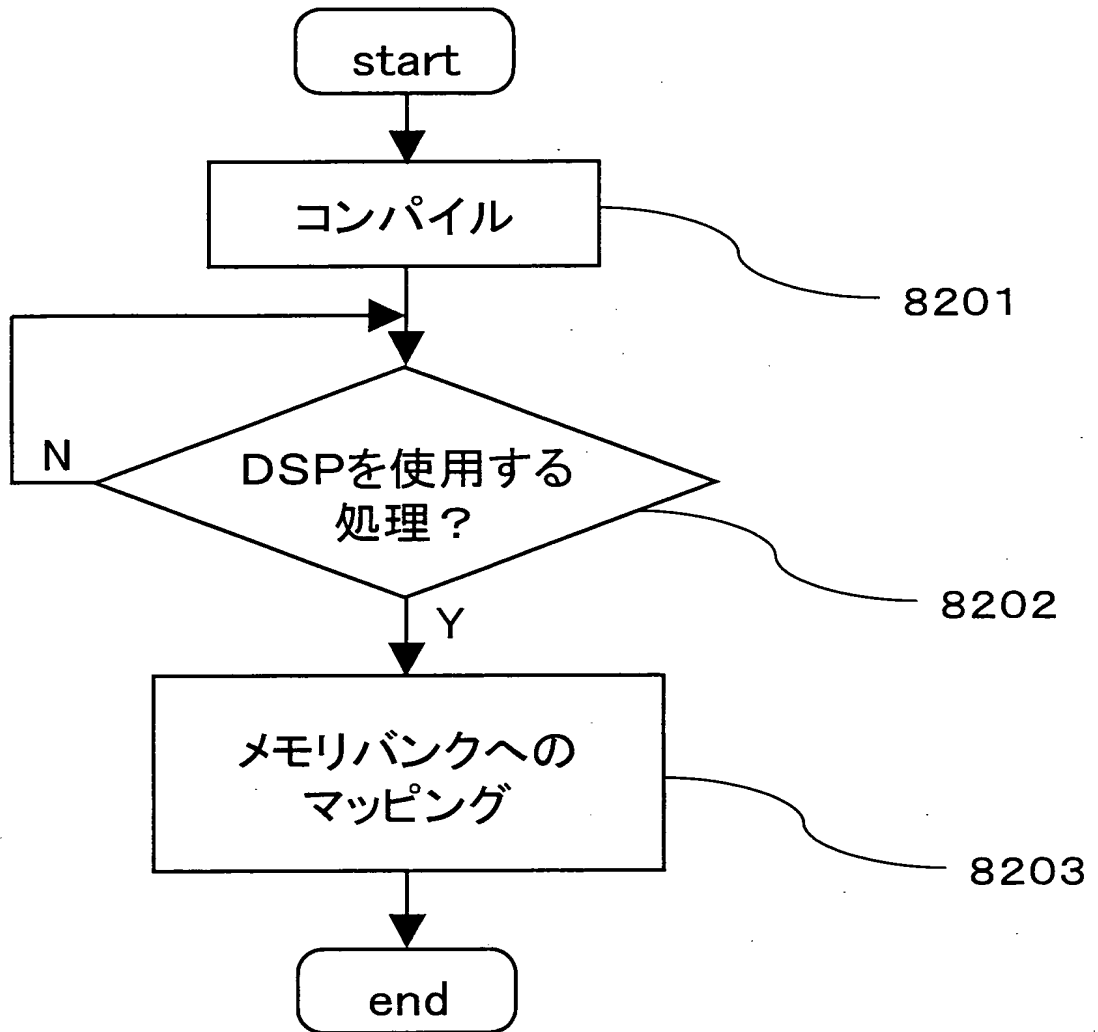




FIG. 36

